ARL-TR-7748 • Aug 2016

US Army Research Laboratory

Improved Solver Settings for 3D Exploding Wire Simulations in ALEGRA

by Robert Doney, Chris Siefert, and John Niederhaus

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TR-7748 ● Aug 2016

US Army Research Laboratory

Improved Solver Settings for 3D Exploding Wire Simulations in ALEGRA

by Robert Doney
Weapons and Materials Research Directorate, ARL

Chris Siefert and John Niederhaus
Sandia National Laboratories, Albuquerque, NM

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* August 2016	**2. REPORT TYPE** Technical Report	**3. DATES COVERED (From - To)** September 2015–August 2016

4. TITLE AND SUBTITLE
Improved Solver Settings for 3D Exploding Wire Simulations in ALEGRA

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Robert Doney, Chris Siefert, and John Niederhaus

5d. PROJECT NUMBER
AH80

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
US Army Research Laboratory
ATTN: RDRL-WMP-D
Aberdeen Proving Ground, MD 21005-5066

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-7748

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the US Department of Energy National Nuclear Security Administration under contract DE-AC04-94AL85000.

14. ABSTRACT

We are interested in simulating a variety of problems in 3 dimensions featuring large electric currents using the Sandia National Laboratories multiphysics code, ALEGRA. However, many such simulations were prohibitively slow due in part to unoptimized (matrix) solver settings. In this report, we address that by varying 6 parameters in the algebraic multigrid preconditioner, Trilinos/ML, as implemented in ALEGRA. Three parameters impacted performance with one dominating. We provide insight to solver behavior as well as recommended values.

15. SUBJECT TERMS
ALEGRA, hydrocode, solvers, multilevel, aztec

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Robert Doney
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	UU	24	**19b. TELEPHONE NUMBER (Include area code)** 410-278-7309

Contents

List of Figures

Approved for public release; distribution is unlimited.

Acknowledgments

This work was made possible by a grant of high-priority computer time (Frontier Project: ARLAP01001YF6) from the Department of Defense (DOD) High Performance Computing Modernization Program at the US Army Research Laboratory (ARL, Excalibur) and the Navy DOD Supercomputing Resource (Haise) Centers. The authors would like to thank the ALEGRA development team for continued code improvements and support, and particularly Rich Drake for assistance with the Dakota implementation. Thanks are extended to Rick Angelini of the ARL Data Analysis and Assessment Center for scripts and tools for visualization software (Ensight) job launching. We thank Drs Carl Kauthauser and James Cazamias for their technical review of this report, and Ms Jessica Schultheis for editorial review.

INTENTIONALLY LEFT BLANK.

1. Introduction

We are interested in simulating a variety of problems in 3 dimensions (3D) featuring large electric currents. While 2D simulations have been quite informative, cylindrical symmetry may interfere with a problem's relevant physics. Specifically, all objects in the domain behave as if they are extruded 360°—turning particles into hoops. In dealing with electrical current, this can have serious ramifications on the current pathways. In 3D (r, ϕ, z) currents can adjust their pathways anywhere along those 360 degrees given the right conditions; however, in 2D (r, z) those pathways can be completely choked off because an insulating hoop, rather than a particle, is present.

Even though our radial trisection mesh is a 90° wedge with just a few million cells, in developing 3D exploding wire benchmarks, we found that simulations were prohibitively expensive. In some cases, it would take a day to advance 1 µs. Given that we require up to approximately 100 µs per calculation, this is clearly a problem.

Users have the ability to modify a large number of solver parameters, some of which can drastically impact performance. However, there is only moderate guidance on what parameters to tune and what they should be. Further, it is expected that they can be problem dependent. Our goal then is to quantify these uncertainties and manually optimize the problem such that we minimize total CPU time without impacting the physics. Only recently did this type of algorithmic uncertainty quantification work become possible through a competed grant of high-priority (Frontier) computing time from the DOD High Performance Computing Modernization Program supporting this and other research areas.

The report is organized as follows. Section 2 describes the computational approach and problem setup while Section 3 provides the results and analysis. Section 4 summarizes the key findings and provides recommendations.

2. Computational Approach

We use several codes and algorithms developed by Sandia National Laboratories (SNL) to evaluate this research. ALEGRA is used to model exploding wires via finite elements using a 90° wedge cylindrical mesh. Simulations were run on Excalibur, a Cray XC40, and for comparison, Haise, an IBM iDataPlex. We found no significant difference in the results. ALEGRA is a finite-element multiphysics simulation code developed by SNL for modeling high-deformation solid dynamics, shock-hydrodynamics, magnetohydrodynamics (MHD), and ferroelectricity.[1–3] It has been used extensively for simulating the large electric currents and the associated heating, material effects, and MHD phenomenology of exploding wires.[4–7] The MHD module of ALEGRA is used here, which incorporates Maxwell's equations into the equations of solid dynamics using the "magneto-quasistatic" approximation.[8] In this approximation, electromagnetic radiation and displacement currents are neglected. Further, although the medium is assumed to have resistivity via Ohm's law, and does carry both electric current and electric and magnetic fields, it is also assumed to be electrically neutral—that is, the electric charge density is zero everywhere, and the medium can have no capacitive behavior. This approximation is sufficient for exploding wires in the regimes considered here, so long as the underlying material models incorporate the effects of ionization on the material electrical conductivity. For exploding wires, the Lee-More-Desjarlais models provide this capability and is known to be reliable through the solid, liquid, gaseous, and warm plasma phases for common metals.[9,10]

In cases where magnetic forces can be neglected, one may use the "low RM" option, which can significantly reduce simulation time.[3,11] The Magnetic Reynold's number, R_M, is derived from the equations of MHD and used to measure the relative importance of competing physical mechanisms. It is defined as $R_M = \mu_0 \sigma L^2/\tau$, where μ_0 is the permeability of free space, σ is the electrical conductivity, and L and τ are the characteristic length and time scales, respectively, of the system. In our case, $\tau(t) \propto 10^{-6} - 10^{-5}$, $\sigma(t) \propto 10^{2} - 10^{6}$, $L(t) \propto 10^{-3} - 10^{-1}$, where several values are time-dependent. At early times, we would expect $R_M \sim 1$. At later times when the wire explodes, this simplified analysis is complicated by the fact that conductivity decreases quickly for the metal plasma—not for the remaining solid wire pieces—and σ can increase for the medium. Each component is important. Additionally, the length scale increases as the plasma expands, and this term may or may not dominate over conductivity. At later time, therefore, we could expect $R_M \sim 0.1 - 10$. In both cases, the low RM option ($R_M << 1$) is not quite applicable and not used in this study.

ALEGRA uses explicit time integration to solve the solid-dynamic and nonresistive component of the multiphysics equations, which are hyperbolic. However, the diffusion of magnetic field and associated resistive heating of the medium are captured in an additional system that is elliptic. These equations are solved implicitly on each timestep using iterative methods and are coupled to the full system via operator splitting.[1] The Trilinos/AztecOO infrastructure is used to form the linear system that represents this problem and solve it using a variety of techniques available to the user via input file controls.

We use the conjugate gradient (CG) solver—an iterative method for solving sparse linear systems—as implemented in Trilinos/AztecOO with a user-specified tolerance on the iterative residual. These may be preconditioned within Trilinos/ML,[12,13] and since we are solving Maxwell's equations, we use the algebraic multigrid preconditioner.[14] This preconditioner has a number of different user-determined settings and parameters. These settings can have a large impact on simulation time, and it is not always clear what values they should have. Accordingly, optimizing 6 of these settings is the primary focus of this report, for which we use Dakota, a toolkit that provides algorithms for optimization, uncertainty quantification, and sensitivity analysis.[15] It can be called directly from within ALEGRA, making it a very attractive tool. In this report we use it primarily to manage the parameter study.

The range of length scale in most exploding wire problems is enormous. A typical wire diameter can be as small as tens or hundreds of microns, but the expanding plasma and shock wave resulting from the wire burst can extend to tens of centimeters. The elliptic nature of the magnetic diffusion problem also demands large computational domains so that boundary conditions are as near to infinitely distant as possible. This results in simulations with many millions of elements or more, and thousands or tens of thousands of timesteps. Parallel computing platforms are needed, and to ensure the large-scale parallel performance of these linear solvers, the multilevel (ML) package from the Trilinos library is used in ALEGRA, coupled to the AztecOO solvers.

In this research, we seek to establish optimal settings for the ML configuration in ALEGRA exploding wire simulations. The ML package uses algebraic multigrid preconditioning to ensure parallel scaling. As such, it is a critical part of ALEGRA-MHD input for determining the overall performance and rate of throughput of ALEGRA-MHD simulations.

The algebraic multigrid algorithms in ML approximate the solution of a linear system on a fine mesh with a series of solutions on progressively coarser meshes.

Figure 1 shows a sample computational mesh, with 2 coarse meshes generated by multigrid. Each coarser mesh represents some set of progressively “lower energy” or “smoother” components of the solution on the fine level solution. By decomposing the solution in this fashion, we can use a basic iterative method, such as Jacobi, Chebyshev, or Gauss-Seidel to rapidly reduce error that is “high energy” on the individual meshes. For the purpose of this study, we will use the Chebyshev smoother on all levels.

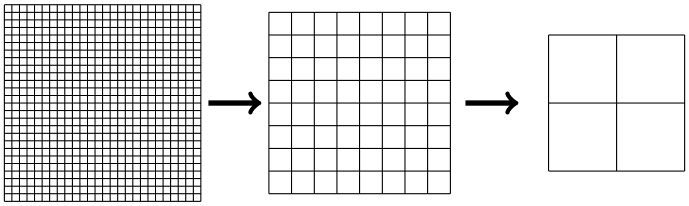

Fig. 1 Example of a multigrid hierarchy, with meshes named A, B, and Z from left to right

Our study will focus on 6 parameters, 3 of which are related to the smoothing process described above, one of which is related to the number of levels, and the final 2 require additional explanation. The first parameter we consider is `FINE SWEEPS`, which represents the number of smoother sweeps used on the user-supplied fine mesh (mesh A in Fig. 1). The second parameter is `INTERMEDIATE SWEEPS`, which represents the number of smoother sweeps used on all meshes except the fine and coarse meshes (all meshes except A and Z in Fig. 1). The third parameter is `COARSE SWEEPS`, which represents the number of smoother sweeps used on the coarsest mesh (mesh Z in Fig. 1). The fourth parameter, `MULTILEVEL GRIDS`, is the maximum number of levels to use. Regardless of the maximum number of levels, ML will stop the coarsening when the coarse grid reaches a certain (sufficiently small) size.

The fifth parameter relates to ML’s ability to coarsen selectively. In the case of stretched meshes, or nonconstant material parameters, the multilevel preconditioner can be more effective by coarsening in particular directions and not in others. In ML, we do this via the `AGGREGATE THRESHOLD` parameter. We use ML’s auxiliary aggregation capability, which means that we “drop” connections in the matrix corresponding to nodes that are sufficiently far away from each other.[13]

The final parameter involves data migration between processors via the `REPARTITION` parameter. This parameter specifies the minimum number of matrix rows to require for each core. If the number of rows per core drops below this level, then cores are idled to keep the overall work-to-communication ratio for the non-idle cores high.

A full factorial of the following ML parameters and their ranges are investigated: `FINE SWEEPS {1,5,10}`, `INTERMEDIATE SWEEPS {1,5,10}`, `COARSE SWEEPS {1,5,10}`, `REPARTITIONING {100,500,1000,2000}`, `MULTILEVEL GRIDS {5,10,15,20,25}`, `AGGREGATE THRESHOLD (AGG) {0.001, 0.005, 0.01, 0.05}`. This consists of 2160 simulations where each case employs roughly 180,000 cells and 33 cores per simulation, yielding approximately 5400 cells per compute core. The parametric study was managed by the Dakota implementation in ALEGRA, and all cases ran to completion.

Each simulation is represented by a 90° radial trisection mesh populated by 2 aluminum plates and an aluminum wire with a sinusoidal surface that oscillates over 25% of its radius. The wire material density is initialized as a random field, sampled from a uniform probability distribution spanning ±1% about a median value of 2704 kg/m^3, with an aggregate, or "grain", size that is twice the radial cell size. This initial density field is stochastic but does not vary from one simulation to the next. Materials are situated in a dry air environment, the latter of which has a temperature ceiling and floor of 50,000 K and 50 K, respectively. For these simulations, we did not perform rigorous convergence work (i.e., mesh resolution, domain size) because the goal was to only find an optimum setting for the parameters investigated. We believe those parameters would not deviate significantly under such convergence analysis. A wire exploding (rather than just heating up) is necessary in the simulation, as we expected the mixed solid-liquid-plasma environment to greatly affect performance. We chose a larger-diameter wire because smaller diameters would require many smaller cells to resolve it. Those smaller cells would further reduce the time step, rendering longer simulation times and becoming very expensive given the scope of this study.

Figure 2 illustrates a sample simulation where the wire is represented by an isovolume of wire temperatures in the solid phase. The wire is colored by current density from a peak value down 6 orders of magnitude. Note in the wire's cross section that its core has a smaller value because current has not had time to diffuse inside further. The wire explosion process has just begun at the smallest diameters of the wire, where the temperatures have exceeded that of melt and those parts of the isovolume are no longer visible. This is also apparent on the clip plane forming a backdrop that plots density in grayscale over roughly 5 orders of magnitude. In time, the event is demarcated by a vertical line in the plots of current and resistance—note the minor inflection in the current trace assisted by a large increase in the resistance since the liquid and gas phase for aluminum has a lower conductivity than as a solid.

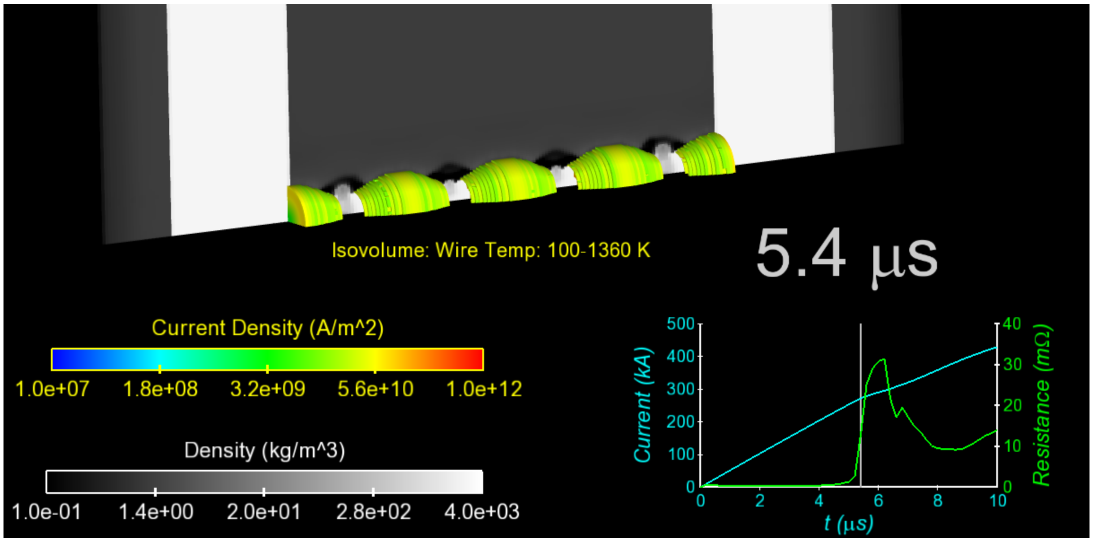

Fig. 2 Sample simulation with onset of wire explosion at 5.4 μs

In these MHD simulations, total CPU time can be greatly reduced by setting a semi-relativisitic limiter, maxfast, to 50 km/s. This limits the maximum velocity of the fast MHD wave speed, which is directly correlated to the time step through the Courant condition. Physically, this is a reasonable choice since the plasma expansion can have radial velocities up to 5 km/s. Our effective value for the speed of the light is therefore much greater than relevant velocities.

Other important syntax that improves simulation stability and runtime is the inclusion of (maximum) temperature clips—which currently require a SESAME equation of state (EOS). Some EOS tables allow for exceedingly high temperatures (10^8 K). Usually when temperatures in the domain exceed 10^5 K, the time step drops precipitously. Rather than allowing this or an errant cell in an extreme state from controlling and possibly crashing the simulation, we can limit the maximum allowable temperature. One must use this with care, however; if the temperature clip is too low, relevant physical processes may be missed. For example, we found that a clip of 25,000 K for all material was over-constrained since increasing it affected the current trace. Also, when viewing temperatures spatially, most of the conducting region was clipping at the maximum value. For this study, we settled on a clip of 75,000 K. For nonelectrified simulations where temperature-induced phase changes are not expected, a high-clip of 3000–5000 K is reasonable. Temperature clips are chosen rather than the more heavy-handed discard option of deleting material from the domain—based on certain triggers—and replacing it with void. This alternate approach violates mass conservation. It also creates a further problem with electromagnetics since void is an insulator and its sudden emergence in large quantity can choke off important current pathways. This problem is more likely to emerge in 2D than 3D, but we still believe that temperature clips are more appropriate here.

It was expected that changing solver parameters should have no effect on the solution. We verified this (not pictured) for several hundred random cases and found the results (e.g., current, inductance, and resistance) to be essentially identical.

3. Results and Discussion

Figure 3 illustrates the results from all 2160 simulations. Each subplot shows the total CPU time as a function of the values for the corresponding ML parameters. Note that spaces between horizontal gridlines are equivalent to roughly 5.5 h. Black dots represent the average for that ensemble of data. The fastest simulation completed in 66,178 s (18.4 h) and the slowest in 156,386 s (43.4 h). Clearly Fine Sweeps has the largest effect on simulation time, with the smallest value being the best performer. Over this full range of inputs, the remaining parameters appear to show little performance trend.

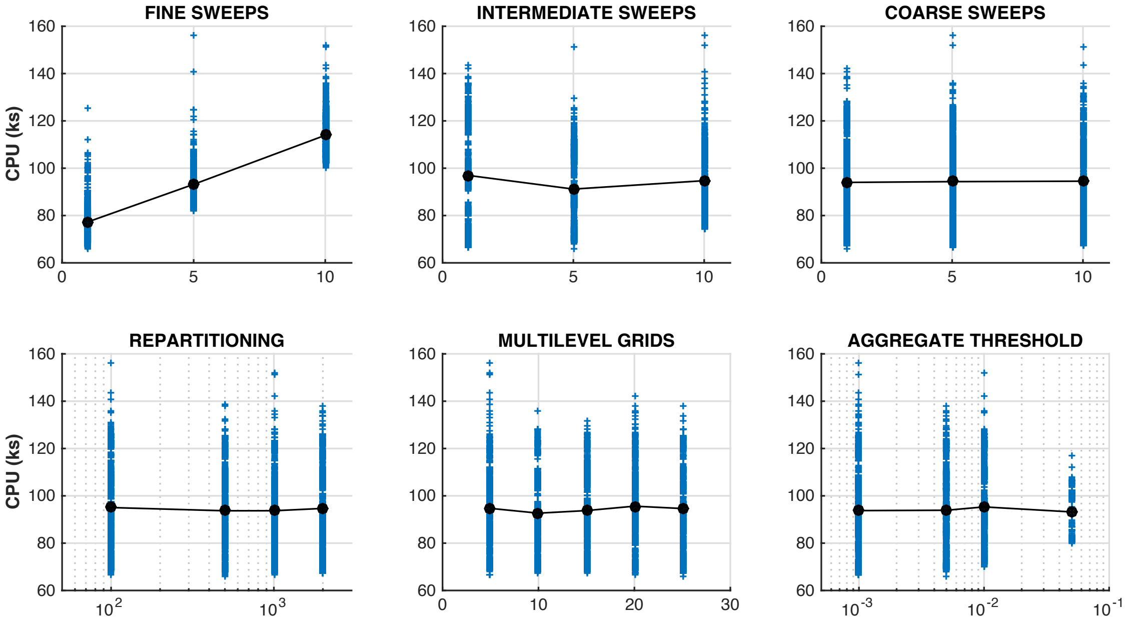

Fig. 3 Timing results from all simulations

Additional performance trends emerge, however, if we look at the fastest simulations. Accordingly, we selected a subset of data—405 cases—where CPU $\leq$ 80 ks (Fig. 4). Fine Sweeps is absent in this figure since it is unity for all of these simulations. Note that spaces between horizontal gridlines are now roughly equivalent to 1.5 h. Finer structure is now apparent in both intermediate sweeps and aggregate threshold, where in both cases the lowest value improves performance when `FINE SWEEPS=1`. There do not appear to be any further significant correlations to performance among the remaining variables in this study.

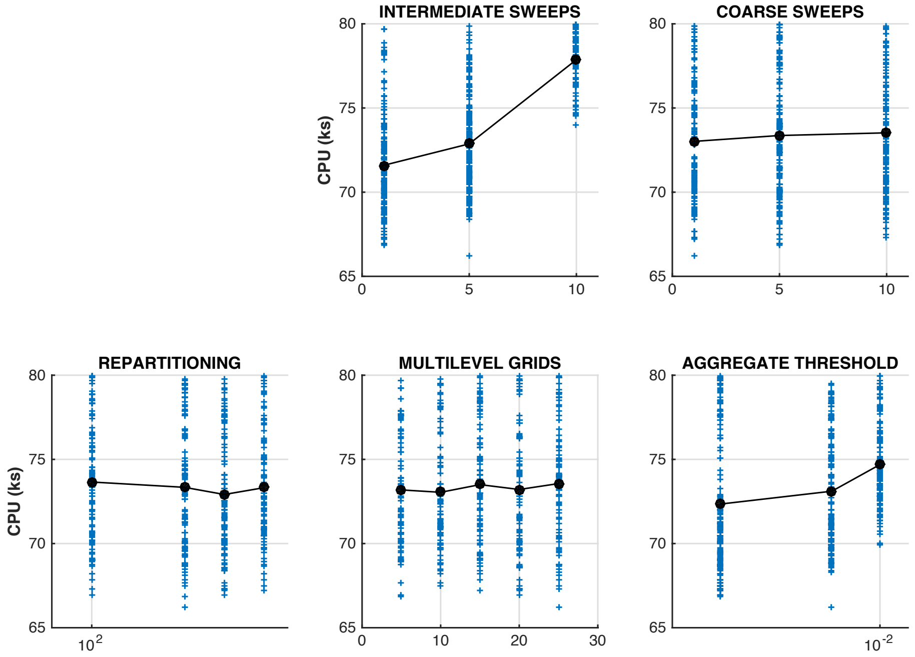

Fig. 4 Timing results from simulations clocked at less than 80 ks

For a selection of data where `AGG=0.001`, we can look at how CPU is affected by the average number of solver iterations per cycle (`ITERSP`) and the average solver solution time per cycle (`SOLTIME`) in a given simulation (Fig. 5). These are user-defined variables rather than normal variable output. In both cases they represent the average value over the required iterations to converge for each cycle. In the left panel, we see a clear correlation between the average solution time and total simulation time. However, in the right panel, islands of data document a surprising result. As the number of iterations increases, the total simulation tends to get smaller. This is counterintuitive as more iterations might be expected to require more overall time. It implies that the cost per iteration can be quite low.

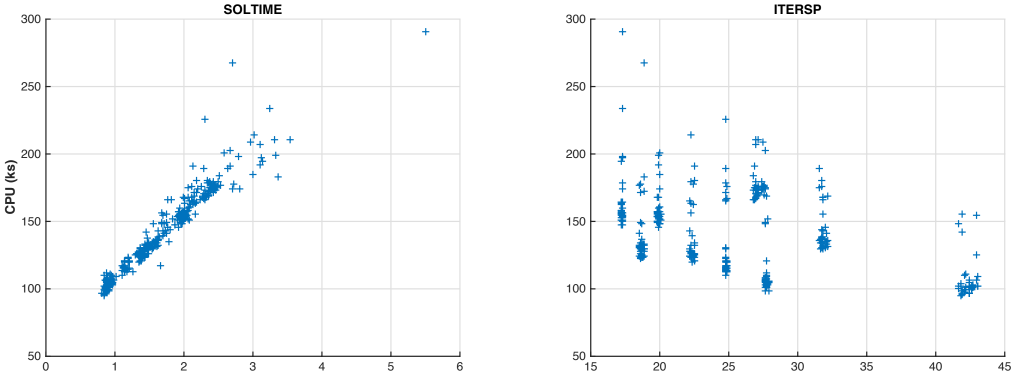

Fig. 5 Effects of average solver solution time and iterations on total solution time

When using the Aztec linear solver, one must always find a balance between long and costly but clever iterations (more sweeps, more levels, etc.) versus quicker, cheaper, but less sophisticated iterations (fewer sweeps, fewer levels). For some problems the former gives faster overall turnaround, and for others, the latter gives faster turnaround. Either way, the problem would be intractable without ML; still, a balance has to be found to minimize the time to solution. The data on the right-hand side of Fig. 5 suggest that for the 3D exploding wire, the latter option is more beneficial: quicker, cheaper iterations. It is unclear what the conditions are for the balance to swing to the other side (i.e., a much smaller wire diameter, or perhaps a mesh with a much larger array of element dimensions and aspect ratios).

4. Conclusion

We performed a 6-parameter study to find the optimum settings in the Trilinos/AztecOO solver for minimizing total CPU time in 3D exploding wire simulations using a 90° radial trisection mesh in ALEGRA. We found the best performance to be most probable when using `FINE SWEEPS=1; INTERMEDIATE SWEEPS=1; AGGREGATE THRESHOLD=0.001`, where the former has the largest effect on simulation time. A separate simulation running with aggregate threshold set at 5.0e-4 showed no further improvements. As expected, changing solver parameters had no effect on the solution. We verified this for several hundred random cases and found the results (e.g., current, inductance, and resistance) to be essentially identical. Our complete input then for the solver is as follows:

```
aztec 1
 solver, cg
 scaling, sym_diag
 conv norm, rhs
 tol= 1.e-6
 max iter = 1000
 output = last
   multilevel
        verbose, 1
        formulation = maxwell2
        fine sweeps = 1
        intermediate sweeps = 1
        coarse sweeps = 1
        multigrid levels = 25
        interpolation algorithm = uc aggregation
        repartition, minimum edges per processor = 500
        smooth prolongator
        aux aggregation
        aggregate threshold 0.001
   end
end
```

Tighter tolerances will of course slow down performance and may be required for certain problems to properly converge. Turning off `AUX AGGREGATION` had no effect on performance time.

Total simulation time for this class of MHD problems can be further improved with the following additions. Dominant among these is `SEMIRELATIVISTIC, MAXFAST`, which limits the fast MHD wave speed. We found that setting it at 5–10 times the maximum physically relevant velocities (i.e., the plasma expansion velocity) greatly improved performance. However, formal convergence analysis is still required to find a minimum acceptable value. Additionally, we recommend that high clips for temperature be used when possible for added stability and speed to prevent an occasional grid cell from reaching cripplingly large temperatures, usually due to mixed materials. Discarding material is also an option once extreme states are reached; however, it introduces void, an insulator, which can alter the current path or even quench it if enough void is present. In these simulations we set $T_{max} = 75,000$ K. A density floor of 0.01 kg/m^3 is a reasonable value and helps prevent the time step from dropping too low. All of these enhancements resulted in up to a 10-fold improvement in total CPU time.

5. References

1. Robinson AC, et al. ALEGRA: an arbitrary Lagrangian-Eulerian multimaterial, multiphysics code. Paper presented at: AIAA 2008. Proceedings of the 46th AIAA Aerospace Sciences Meeting and Exhibit; 2008 Jan 7-10; Reno, NV. AIAA Paper No.: AIAA-2008-1235.

2. Robinson AC, et al. ALEGRA user manual. Albuquerque (NM): Sandia National Laboratories; 2015. Report No.: SAND2015-4970.

3. Robinson AC, et al. ALEGRA-MHD user manual. Albuquerque (NM): Sandia National Laboratories; 2015. Report No.: SAND2015-4971.

4. Doney R, Vunni G, Niederhaus J. Experiments and simulations of exploding aluminum wires: validation of ALEGRA-MHD. Aberdeen Proving Ground (MD): Army Research Laboratory; 2010. Report No.: ARL-TR-5299.

5. Drake R, Niederhaus J, Koenig D. A parametric study in MHD modeling for exploding wires using Dakota in ALEGRA. Aberdeen Proving Ground (MD): 2012 May; https://cfwebprod.sandia.gov/cfdocs/CompResearch/docs/ExplodingWireUUR_paper.pdf.

6. O'Malley P, Garasi C. Understanding the electrical interplay between a capacitive discharge circuit and exploding metal. Albuquerque (NM): Sandia National Laboratories; 2015. Report No.: SAND2015-1132.

7. Ashcraft C, Niederhaus J, Robinson A. Verification and validation of a coordinate-transformation method in axisymmetric transient magnetics. Albuquerque (NM): Sandia National Laboratories; 2016. Report No.: SAND2016-0804.

8. Melcher J. Continuum electromechanics. Cambridge (MA): MIT Press; 1968. Also available online from MIT OpenCourseWare at http://ocw.mit.edu.

9. Desjarlais M. Practical improvements to the Lee-More conductivity near the metal-insulator transition. Contributions to Plasma Physics. 2001;41(2–3):267–270.

10. Desjarlais M, Kress J, Collins L. Electrical conductivity for warm, dense aluminum plasmas and liquids. Phys Rev E. 2002;66(2):025401(R).

11. Garasi CJ, Sandia National Laboratories, Albuquerque, NM. Personal communication, 2016.

12. Heroux M, et al. An overview of Trilinos. Albuquerque (NM): Sandia National Laboratories; 2003. Report No.: SAND2003-2927.

13. Gee M, Siefert C, Hu J, Tuminaro R, Sala M. ML 5.0 smoothed aggregation user's guide. Albuquerque (NM): Sandia National Laboratories; 2006. Report No.: SAND2006-2649.

14. Bochev P, Hu J, Siefert C, Tuminaro R. An algebraic multigrid approach based on compatible gauge reformulations of Maxwell's equations. SIAM J Sci Comput. 2008;31(1):557–583.

15. Adams B, et al. A multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: version 6.3 user's manual. Albuquerque (NM): Sandia National Laboratories; 2014. Report No.: SAND2014-4633.

List of Symbols, Abbreviations, and Acronyms

1-2-3D	1-2-3-dimensional
CG	conjugate gradient
CPU	central processing unit
EOS	equation of state
MHD	magnetohydrodynamics
ML	multilevel
SNL	Sandia National Laboratories

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

2 DIRECTOR
(PDF) US ARMY RESEARCH LAB
RDRL CIO L
IMAL HRA MAIL & RECORDS MGMT

1 GOVT PRINTG OFC
(PDF) A MALHOTRA

10 SANDIA NATIONAL LABORATORIES
(PDF) K COCHRANE
C GARASI
G HANSEN
R KRAMER
D LABRECHE
J NIEDERHAUS
S PETNEY
A ROBINSON
J SANCHEZ
E STRACK

1 PEO GCS
(PDF) J ROWE

3 ENIG ASSOCIATES
(PDF) M BARNARD
D BENTZ
E ENIG

1 INL
(PDF) J LACY

ABERDEEN PROVING GROUND

12 DIR USARL
(PDF) RDRL WML H
B AYDELOTTE
RDRL WMP A
M COPPINGER
C UHLIG
J CAZAMIAS
RDRL WMP C
R BECKER
RDRL WMP D
J RUNYEON
G VUNNI
M ZELLNER
RDRL WMP E
D HORNBAKER
J HOUSKAMP
RDRL WMP F
J NESTA
J PRITCHETT